

UNITED STATES PATENT APPLICATION

for

IMAGE PROCESSING APPARATUS, IMAGE PROCESSING METHOD,  
AND STORAGE MEDIUM

Inventors:

Takanori Yano

Takao Inoue

Akira Takahashi

Keiichi Ikebe

Hiroyuki Sakuyama

Taku Kodama

Ikuko Yamashiro

Takashi Maki

prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP

12400 Wilshire Boulevard

Los Angeles, CA 90025-1026

(408) 720-8300

File No.: 006453.P007

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number: EV 336582235 US

Date of Deposit: September 5, 2003

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to Mail Stop Patent Application, Commissioner for Patents, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Debra L. Rigglo

(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

(Date signed)

**IMAGE PROCESSING APPARATUS, IMAGE PROCESSING METHOD,  
AND STORAGE MEDIUM**

**[0001]** The present application claims priority to the corresponding Japanese Application No. 2002-261739 filed on September 6, 2002, the entire contents of which are hereby incorporated by reference.

**BACKGROUND OF THE INVENTION**

**Field of the Invention**

**[0002]** The present invention relates to an image processing apparatus for a static (still) image or a moving (motion) image, and particularly to an image processing apparatus that processes encoded data of an image in an encoded state.

**Description of the Related Art**

**[0003]** In the case of a digital camera, a digital video camera, a monitoring camera, and a monitoring system, for example, data of an image acquired by such a camera is generally compressed and encoded, and then is stored in a recording medium (or a storage medium). JPEG or MPEG is widely employed as an image compression and encoding method. In a case of JPEG or MPEG, when compressed and encoded image data is further compressed, the compressed and encoded image data is temporarily decoded and expanded (or decompressed at first), and is then compressed and encoded again.

**[0004]** A compression rate of the compression and encoding process is previously set by taking into account the capacity of a storage medium that stores encoded data. For example, according to Japanese Patent Application Laid-Open No. 5-64001, when the size of encoded data is determined to be larger than a memory size, an image processing apparatus decodes and expands the encoded data, and

compresses the reproduced image data at a higher compression ratio again.

[0005] Recently, JPEG2000 (ISO/IEC FCD 15444-1) and Motion-JPEG2000 (ISO/IEC FCD 15444-3) are widely regarded as a new compression and encoding method that can replace JPEG or MPEG. In the case of JPEG, image degradation occurs at the time of the recompressing due to a generation (a compression type). On the other hand, in a case of data encoded by JPEG2000, a code amount (a compressing rate) can be adjusted in an encoded state, and image degradation due to a generation such as in JPEG does not occur. In the case of Motion-JPEG2000, consecutive plural static images constituting a moving image are handled as respective frames, but encoded data of each frame follows JPEG2000. A detailed description of JPEG2000 is found in "Next Generation Image Encoding Method JPEG2000" (author: Yasuyuki Nomizu, published by Triceps Inc., on February 13, 2001).

[0006] When the remaining capacity of a storage medium that stores encoded image data becomes insufficient, there is a case where it is required to decrease the code amount of the encoded data. In this case, in order to determine an appropriate decreasing code amount by taking into account image quality, it is preferable to perform code amount decreasing over again from scratch. However, it is desired to prevent the image quality degradation at the time of the recompression due to a generation.

[0007] In the case of determining the contents of the moving image, it is efficient to decrease the number of frames to be reproduced in order to decrease the reproducing time. Particularly, in the case of a moving image acquired by the monitoring camera, if only the frames that include a target person or only the frames that correspond to a target motion can be reproduced, it is possible to efficiently

determine the contents of the moving image. Furthermore, in order to determine the contents of the moving image in detail, it is preferable to also reproduce images of the frames that were not reproduced in accordance with a necessity. Meanwhile, it is convenient to appropriately discard frames that are determined as frames having less important contents.

## **SUMMARY OF THE INVENTION**

**[0008]** An image processing method, apparatus and article of manufacture are described. In one embodiment, the image processing apparatus comprises a partial codestream truncating unit to partially truncate ordinarily encoded data of an image to create a truncated codestream, in accordance with a selectively truncation command, and a restoration unit to, in accordance with an undo command, restore the original encoded data from which the truncated code stream was created temporarily by the partial codestream truncating unit.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

- [0009] FIG. 1 is a block diagram showing a flow diagram of JPEG2000;
- [0010] FIGS. 2A through 2D show a two-dimensional wavelet transformation in a case where the decomposition level number is three;
- [0011] FIG. 3 shows a format of encoded data of JPEG2000;
- [0012] FIG. 4 is an illustration of a precinct and a code block;
- [0013] FIG. 5 shows an example of division of a packet and a layer;
- [0014] FIG. 6 shows an example of an arranging order and an interpreting order of packets in a case of an LRCP progression order;
- [0015] FIG. 7 shows an exemplary configuration of an image processing apparatus according to one embodiment of the present invention;
- [0016] FIG. 8 is a flowchart showing an operation mode that includes performing a partial codestream truncation process, a restoration process, and a code line discarding process for individual images;
- [0017] FIG. 9 is an illustration of a code line temporary truncation process;
- [0018] FIG. 10 is an illustration of a restoration process;
- [0019] FIG. 11 is a flowchart showing an operation mode that includes successively performing the code line temporary truncation process for frames of a moving image;
- [0020] FIG. 12 is a flowchart showing an operation mode that includes temporarily cutting, successively, frames by the frame of a moving image;
- [0021] FIG. 13 is an illustration of the frame-by-frame temporary code line cutting and the restoration process of the moving image;
- [0022] FIGS. 14A through 14D show a relationship between camera vibration and a high-frequency component amount in each direction;

- [0023] FIG. 15 shows a configuration of a main header;
- [0024] FIGS. 16A and 16B show a configuration of a tile header;
- [0025] FIG. 17 is a table of a marker and a marker segment;
- [0026] FIG. 18 shows a configuration of an SIZ marker segment;
- [0027] FIG. 19 shows a configuration of a COD marker segment;
- [0028] FIG. 20 shows a configuration of a COC marker segment;
- [0029] FIG. 21 shows a configuration of a QCD marker segment;
- [0030] FIG. 22 shows a configuration of a QCC marker segment;
- [0031] FIG. 23 shows a configuration of a COM marker segment; and
- [0032] FIG. 24 is a flowchart showing an operation mode that includes reproducing a moving image that has undergone a frame-by-frame code line temporary truncation process.

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

**[0033]** One embodiment of the present invention provides a novel image processing apparatus and image processing method to realize one or more of the desires set forth in the Description of the Related Art section above.

**[0034]** In one embodiment of the present invention, an image processing apparatus comprises a partial codestream truncating unit for partially truncating some form of ordinarily encoded data of an image to create a truncated codestream, in accordance with a selectively truncation command, and a restoring unit for, in accordance with an undo command, restoring the original encoded data from which the truncated code stream was created temporarily by the partial codestream truncating unit.

**[0035]** According to another embodiment of the present invention, the partial codestream truncating unit truncates ordinarily encoded data by turning the original encoded data into second encoded data, causing the truncated codestream to be stored, whereby the second encoded data does not include the truncated codestream, and the restoring unit combines the second encoded data and the stored truncated codestream to restore the original encoded data.

**[0036]** According to another embodiment of the present invention, the partial codestream truncating unit changes original tag information of the original encoded data to second tag information indicating that the truncated codestream is not subjected to decoding, and the restoring unit turns the second tag information back into the original tag information to restore the original encoded data.

**[0037]** According to another embodiment of the present invention, the image is a moving image having a group of frames, the partial codestream truncating unit temporarily truncates, as the truncated codestream, the entire encoded data of at least



one frame of the moving image from the group of frames of the moving image to cause the entire encoded data of the at least one frame to be stored, and the restoring unit includes the entire encoded data of the at least one frame in the group of frames of the moving image to restore the original encoded data.

**[0038]** According to another embodiment of the present invention, the image is a moving image having frames, the partial codestream truncating unit adds, to the entire encoded data of at least one frame of the moving image, information indicating that the entire encoded data as the truncated codestream is not subjected to decoding, and the restoring unit discards the information added to the encoded data of the at least one frame to restore the original encoded data.

**[0039]** According to another embodiment of the present invention, encoded data of an image processed by the image processing apparatus or the image processing method is based on JPEG2000 or Motion-JPEG2000, or is based on a format similar to JPEG2000 or Motion-JPEG2000 in which data can be processed in an encoded state.

**[0040]** Other objects, features, and advantages of the present invention will become more apparent from the following detailed description when read in conjunction with the accompanying drawings.

**[0041]** Embodiments of the present invention will now be described in conjunction with the several Figures. In the following embodiments, encoded data of a static image may be based on JPEG2000 (ISO/IEC FCD 15444-1), and encoded data of a moving image may be based on Motion-JPEG2000 (ISO/IEC FCD 15444-3).

**[0042]** An algorithm of JPEG2000 is described in detail in the above-mentioned document, but the algorithm of JPEG2000 is described within a range suitable for describing the following embodiments.

**[0043]** FIG. 1 shows a simplified block diagram for the algorithm of JPEG2000. Each component of the image data that is to be compressed and encoded (in a case of handling a moving image, each component of image data corresponding to each frame) is divided into tiles. The tiles are rectangular regions that do not overlap. The image data is processed by the tile per component. It should be noted that the size of the tile can be made the same size as that of the image, i.e., the process can be performed without dividing the image data into tiles.

**[0044]** In order to improve a compressing ratio, color space conversion from RGB data or CMY data into YCrCb data is performed on a tile image at Step S1 as shown in FIG. 1. However, this color space conversion may be omitted.

**[0045]** At Step S2, two-dimensional wavelet transformation (disperse wavelet transformation DWT) is performed on each tile image of each component after the color space conversion.

**[0046]** FIGS. 2A through 2D are illustrations of a case in which the decomposition level number is three. The two-dimensional wavelet transformation is performed on a tile image (shown in FIG. 2A) having the decomposition level of zero so that the tile image of FIG. 2A can be divided into sub-bands 1LL, 1HL, 1LH and 1HH, as shown in FIG. 2B. The two-dimensional wavelet transformation is applied to a coefficient of the sub-band 1LL so that the sub-band 1LL can be divided into sub-bands 2LL, 2HL, 2LH and 2HH as shown in FIG. 2C. The two-dimensional wavelet transformation is applied to a coefficient of the sub-band 2LL so that the sub-band 2LL can be divided into sub-bands 3LL, 3HL, 3LH and 3HH as shown in FIG. 2D.

**[0047]** The wavelet coefficients obtained by such octave division of the low frequency component (coefficient of the sub-band LL) is quantized for each sub-band at Step S3. In JPEG2000, it is possible to perform either lossless (reversible)

compression or lossy (nonreversible) compression. Further, in JPEG2000, in the case of the lossless compression, the quantizing step width is always “1” so that the quantizing is not performed.

**[0048]** After the quantizing is performed, the coefficient of each sub-band is entropy-encoded at Step S4. In one embodiment, the entropy encoding method is called an EBCOT (Embedded Block Coding with Optimized Truncation) that includes block division, coefficient modeling, and binary arithmetic encoding. In the entropy encoding, a bit-plane of the coefficient of each sub-band after quantizing is encoded from an upper plane to a lower plane by the block called a code block.

**[0049]** The last two steps S5 and S6 are code forming processes. At step S5, the codes created at Step S4 are collected to form a packet. Next, at Step S6, the packets created at Step S5 are arranged in accordance with a progression order, and necessary tag information is added to the arranged packets to form encoded data in a predetermined format. In JPEG2000, five types of progression orders based on combination of a resolution level, a position, a layer, and a component (color component) are defined for encoding order control.

**[0050]** FIG. 3 shows a format of the JPEG2000 encoded data. As shown in FIG. 3, the encoded data begins with a tag called an SOC marker that indicates the start of the encoded data. Tag information called a main header follows the SOC marker. The main header describes an encoding parameter, a quantizing parameter, and so on. After the main header, the code data of each tile begins with a tag called an SOT marker. The code data of each tile includes tag information called a tile header, a tag called an SOD marker, and tile data having the contents of a code line of each tile. After the last tile data, a tag called an EOC marker is placed for indicating the end. In this specification, the term “code line” means a part of codestream.

**[0051]** According to the JPEG2000 algorithm, image quality is good at a high compression rate (a low bit rate). Such a JPEG2000 algorithm has many characteristics.

**[0052]** As for one characteristic of the JPEG2000 algorithm, it is possible to adjust an entire code amount without performing recompression by performing post-quantization such that the code of the encoded data is truncated (truncated). This code truncation can be performed in various units such as a tile region, a component, a decomposition level (resolution level), a bit plane, a sub-bit plane, a packet, and a layer (in the case of multi-layers). The relation between the decomposition level and the resolution level is shown in FIG. 2D in the way that the numbers in the parentheses indicate the resolution levels of the corresponding sub-bands. As for another characteristic of the JPEG algorithm, encoded data can be easily divided into two or more encoded data in the encoded state, and these encoded data can be combined to restore the original encoded data. As for still another characteristic of the JPEG2000 algorithm, by only rewriting (changing) tag information of encoded data, decoding can be performed as if a part of the code line is actually truncated.

**[0053]** Next, a precinct, a code block, a packet and a layer will be briefly described. There is the following size relation: image size  $\geq$  tile size  $\geq$  sub-band size  $\geq$  precinct size  $\geq$  code block size.

**[0054]** A precinct refers to a rectangular region of the sub-bands. As shown in FIG. 4, combination of three regions that are located at the spatially same positions of the sub-bands HL, LH and HH having the same decomposition level is treated as one precinct. However, as for the sub-band LL, one region is treated as one precinct. The size of the precinct can be made equal to the size of the sub-band. A code block is a rectangular region that is created by dividing the precinct. For simplicity, the

precinct and one code block at the decomposition level 1 are shown in FIG. 4.

[0055] A packet is created by picking up and collecting a part of codes of the all code blocks included in the precinct (for example, the codes of three bit planes from the uppermost bit plane to the third bit plane). A packet having an empty code can be allowed. Thus, the codes of the code blocks are collected to create packets, and the packets are arranged in accordance with a desired progression order to form encoded data. In FIG. 3, tile data (a part for each tile) after the SOD marker corresponds to the gathered packets.

[0056] By collecting the packets of the all precincts (i.e., all code blocks, all sub-bands), a part of the codes of the entire image region (for example, this part corresponds to the codes of the uppermost bit plane to the third bit plane of the wavelet coefficients of the entire image region) is created as a layer. However, as shown in the following example, the layer does not necessarily have to include the packets of all precincts. Accordingly, when a larger number of layers are decoded at the time of expanding, the image quality of the reproduced image is more improved. In other words, a layer can be considered as a unit of the image quality. By collecting all layers, the codes of all bit planes of the entire image region are made.

[0057] FIG. 5 shows an example of the packets and layers in a case where the decomposition level number is 2 (the resolution level number is 3). In FIG. 5, rectangles having longer lengths in a vertical direction are packets, and the numbers shown in these rectangles are the packet numbers. Layers correspond to rectangles defined with the dark and light coloring, respectively. That is, the codes of the packets having the packet numbers 0 through 16 constitute the layer 0, the codes of the packets having the packet numbers 17 through 33 constitute the layer 1, the codes of the packets having the packet numbers 34 through 50 constitute the layer 2, the

codes of the packets having the packet numbers 51 through 67 constitute the layer 3, the codes of the packets having the packet numbers 68 through 84 constitute the layer 4, the codes of the packets having the packet numbers 85 through 101 constitute the layer 5, the codes of the packets having the packet numbers 102 through 118 constitute the layer 6, the codes of the packets having the packet numbers 119 through 135 constitute the layer 7, the codes of the packets having the packet numbers 136 through 148 constitute the layer 8, the codes of the packets having the packet numbers 149 through 161 constitute the layer 9. The corresponding relation between packets and precincts changes depending on a different progression order and the number of divided layers. The layer configuration shown in FIG. 5 is only one example.

[0058] JPEG2000 has another characteristic in which a progression order of encoded data can be changed in the encoded state. In JPEG2000, five progression orders LRCP, RLCP, RPCL, PCRL and CPRL are defined where L designates a layer, R a resolution level, G a component, and P a precinct (position).

[0059] In a case of the LRCP progression, the packet arrangement order (at the time of the encoding) or the packet interpretation order (at the time of the decoding) is represented as the following for-loop nested in the order of L, R, C, P.

```
for (layer) {  
  for (resolution level) {  
    for (component) {  
      for (precinct) {  
        arrange packets: when encoding  
        interpret packets: when decoding  
      }  
    }  
  }  
}
```

} .

**[0060]** In a specific example, the image size is  $100 \times 100$  pixels (without tile dividing), the number of layers is 2, the resolution level is 3 (levels 0 through 2), and the component number is 3, a precinct size is  $32 \times 32$ . In this example, 36 packets are arranged and interpreted in the manner shown in FIG. 6.

**[0061]** Furthermore, in a case of the RLCP progression, the packet arrangement (at the time of the encoding) or the packet interpretation (at the time of the decoding) is performed in the following order:

```
for (resolution level) {  
  for (layer) {  
    for (component) {  
      for (precinct) {  
        arrange packets: when encoding  
        interpret packets: when decoding  
      }  
    }  
  }  
}
```

**[0062]** Also in cases of the other progression orders, a packet arranging order or a packet interpreting order can be determined by the nested for-loop.

**[0063]** Tag information and rewriting (changing) of the tag information will be described later.

**[0064]** In the following, one embodiment of the present invention will be described.

**[0065]** FIG. 7 is a block diagram of one embodiment of an image processing apparatus. An image processing apparatus having a configuration shown in FIG. 7

includes an interface unit 1000 for inputting encoded data of a static image or a moving image. In one embodiment, the interface unit 1000 directly takes in the encoded data from equipment such as an external personal computer via a wired or wireless transmission line or a network. The image processing apparatus includes another interface unit 1001 that takes in image data from equipment such as an external personal computer via a wired or wireless transmission line or a network. The image processing apparatus further includes an image acquiring unit 1002 (for example, a monitoring camera) that inputs image data of a static image or a moving image by recording a shooting object.

[0066] The image processing apparatus further includes an image compressing unit 1003 that is based on JPEG2000 and compresses and encodes the image data input from the interface unit 1001 or the image acquiring unit 1002 to generate encoded data. However, the image processing apparatus may be configured to include only the interface unit 1000. In another embodiment, the image processing apparatus may be configured not to include the image acquiring unit 1002 or not to include the interface unit 1001. Furthermore, according to one embodiment of the present invention, the interface unit 1000 may be replaced with a unit for reading encoded data from a storage medium such a memory card. The image processing apparatus further includes a storage unit (A) 1004A that stores the input encoded data. The image processing apparatus may include other units as shown in FIG. 7.

[0067] In the image processing apparatus, a code line temporary truncation process unit 1010 performs the code line temporary deletion process under the control of a control unit 1020. In accordance with a code line truncation command (selectively truncation command), the code line temporary truncation process unit 1010 reads from the storage unit (A) 1004A input encoded data or encoded data



restored by a restoration process, truncates a truncating target code line (a part of the original code line) from the original code line, creates second encoded data in which the truncating target code line is truncated from the original code line), overwrites the second encoded data of the original encoded data on the original encoded data stored in the storage unit (A) 1004A, and stores the truncated code line in a storage unit (B) 1004B. The storage unit (A) 1004A and the storage unit (B) 1004B do not need to be physically independent of each other, and may be assigned to different storing regions, respectively, in the same storage unit. The truncated code line may be stored in any form, but in the first embodiment, the code line temporary truncation process unit 1010 creates the encoded data having only the truncated code line in a JPEG2000 format for the restoration process, and the storage unit (B) 1004B stores this encoded data created by the code line temporary truncation process unit 1010. The code line temporary truncation process unit 1010 also can perform the code line deletion (temporary culling of the frames) by the frame (by the image) for a moving image (i.e., by the unit of a frame). In this case, original encoded data of frames of the moving image is stored in the storage unit (B) 1004B, and the original encoded data of the frames stored in the storage unit (A) 1004A is truncated. That is, these frames are actually truncated from a group of frames of the moving image.

**[0068]** A restoration process unit 1011 of the image processing apparatus functions to, under the control of the control unit 1020, restore encoded data that is data before the code line temporary truncation process is performed. In accordance with an undo command, the restoration process unit 1011 reads the encoded data stored in the storage unit (A) 1004A on which the code line temporary truncating has been performed, and the encoded data that consists of the truncated code line stored in the storage unit (B) 1004B. The restoration process unit then combines the

encoded data that has undergone the code line temporary truncation process with the encoded data that consists of the truncated code line to restore the original (input) encoded data. The restored encoded data is overwritten on the encoded data in the storage unit (A) 1004A on which the code line temporary truncation process has been performed, and the encoded data that consists of the truncated code line is erased from the storage unit (B) 1004B. In a case where frames of a moving image are temporarily cut by the code line temporary truncation process, the encoded data of the frames restored by the restoration process unit is simply written in the storage unit (A) 1004A, and the encoded data of the temporarily cut (truncated) frames stored in the storage unit (B) 1004B is only erased. This means that the temporarily cut frames are made to return to the group of frames of the moving image.

**[0069]** A code line discarding process unit 1012 functions to discard the temporarily truncated code line under the control of the control unit 1020. Under the control of the control unit 1020, the code line discarding process unit 1012 erases the encoded data stored in the storage unit (B) 1004B that consists of the encoded code line truncated by the code line temporary truncation process. After the code line discarding process is performed, it becomes impossible to restore the original encoded data that includes the discarded code line. In a case of the temporarily cut frames, after the code line discarding process is performed, it becomes impossible to restore the discarded frames.

**[0070]** An image expanding unit 1013 functions to decode and expand the encoded data stored in the storage unit (A) 1004A in accordance with the algorithm of JPEG2000 (or Motion-JPEG2000) to reproduce the image data. The reproduced image data is displayed as an image by a displaying unit 1014. The displaying unit 1014 is used not only for displaying the reproduced image but also for image

selecting or information exchanging with a user.

**[0071]** The control unit 1020 functions to control an operation of the image processing apparatus and/or an operation of each unit of the image processing apparatus. An operation unit 1021 is input units that are operated by the user, and is used for inputting to the control unit 1020 an image selecting instruction, a code line truncating instruction, a code line truncating method (a unit of truncating), an undo instruction, an undo releasing instruction, and so on. An analyzing unit 1022 functions to analyze the encoded data to obtain information for selecting frames to be temporarily cut. The control unit 1020 issues a code line truncation command, an undo command, and an undo releasing command in accordance with the information obtained by the analyzing of the analyzing unit 1022.

**[0072]** The code line temporally truncation process unit 1010, the restoration unit 1011, the code line discarding process unit 1012, the analyzing unit 1022, and so on may be realized as hardware, but may be realized as software by using a general-purpose computer such as a personal computer, or a special computer such as a micro computer. Similarly, the control unit 1020, the image compressing unit 1003, and the image expanding unit 1013 may be realized as hardware and/or software. Further, these units may be realized as combination of hardware and software. The scope of the present invention includes programs for these units, and a computer-readable storing (recording) medium that stores the programs. Such scope of the present invention applies to a second embodiment (described later) of the present invention.

**[0073]** Next, an operation of the image processing apparatus will be described.

**[0074]** First, an operation mode (1-1) in which a user specifies encoded data of one image (for example, one static image or one frame of a moving image), and the code line truncating of the specified encoded data is performed while the user can

confirm a result of the code line truncating. FIG. 8 shows a flowchart of this operation mode (1-1).

**[0075]** When an image (for example, one frame of a moving image) is specified, encoded data of the specified image is read from the storage unit (A) 1004A, and decoded and expanded by the image expanding unit 1013 so that the reproduced image data can be displayed on the displaying unit 1014 at Step S101.

**[0076]** When the user determines a necessity or desire to perform the code line truncating by viewing the displayed image, the user specifies a truncating method, and inputs the code line temporary truncating instruction. In one embodiment, specifying the code line truncating method means specifying a unit to be truncated. Specifically, for example, specifying the code line truncating method means specifying code line truncating by the layer (the number of layers to be truncated), code line truncating by the resolution level (the resolution level number to be truncated), code line truncating by the bit plane (the bit plane number to be truncated), or code line truncating by the component (a component to be truncated). The user can select a code line truncating unit (e.g., a layer, the resolution level number, and so on) by inputting an instruction indication such a unit.

**[0077]** The control unit 1020 determines the contents of the input from the user at Step S102, and performs control in accordance with the contents of the input.

**[0078]** When the input from the user is the code line truncating instruction, the code line truncation command that includes information about the code line truncating method specified by the user is issued from the control unit 1020 to the code line temporary truncation process unit 1010 so that the code line temporary truncation process can be performed on the encoded data in accordance with the code line truncating method specified by the user under the control of the control unit 1020.

Next, under the control of the control unit 1020, the encoded data in which the truncating target code line has been truncated by the code line temporary truncation process is read from the storage unit (A) 1004A, and is decoded and expanded by the image expanding unit 1013, and the decoded and expanded image data is displayed on the displaying unit 1014 at Step S104.

**[0079]** The user can confirm quality of the displayed image. When the user wants to perform the code line truncation process over again from scratch, the user can input an undo instruction. On the other hand, when the user wants to fix the result of the current code line truncation process, the user can input an undo releasing instruction. Further, when the user wants to end the operation, the user can input the end instruction.

**[0080]** When an undo instruction is input, an undo command is issued from the control unit 1020 to the restoration process unit 1011, and the restoration process is performed by the restoration process unit 1011 at Step S105. Then, under the control of the control unit 1020, the restored encoded data (i.e., the original encoded data before the code line truncation process is performed) is read from the storage unit (A) 1004A, and is decoded and expanded in the image expanding unit 1013, and the decoded and expanded image is displayed on the displaying unit 1014 at Step S106. The user can input the code line truncation command again with a different code truncating method for the encoded data.

**[0081]** In this manner, the user can repeatedly cut a code amount of the encoded data. Since the encoded data is not recompressed, the image degradation caused by a generation does not occur.

**[0082]** When an undo releasing instruction is input from the user, an undo releasing command is issued from the control unit 1020 to the code line discarding

process unit 1012 where the code line discarding process is performed at Step S107.

In other words, the encoded data stored in the storage unit (B) 1004B that consists of the truncating target code line truncated from the original code line is erased.

Accordingly, the original encoded data that has the truncating target code line can be no longer restored.

**[0083]** When the code amount of the encoded data is cut for shortage of an available capacity of the storage unit (A) 1004A, the code line discarding process is performed so that the available capacity of the storage unit (A) 1004A can be increased.

**[0084]** After the code line temporary truncation process is performed, the end instruction may be input to end the operation. However, at another time point, this operation mode (1-1) may be used again, and an undo command or an undo releasing command may be input for the encoded data for which the operation was ended so that the restoration process or the code line discarding process can be performed.

**[0085]** FIGS. 9 and 10 are illustrations plainly showing an exemplary code line temporary truncation process that is performed by the tile, and the restoration process, respectively. As shown in FIG. 9, the code line temporary truncation process converts encoded data 2000 of an image 2001 into encoded data 2002 of an image 2003 such that four tiles at the lower half part of the image 2001 are truncated in the image 2003. A code line of an image 2005 having the four tiles of the lower half part of the image 2001 is stored as encoded data 2004. When an undo instruction is input, as shown in FIG. 10, the restoration process combines the encoded data 2002 and the encoded data 2004 to restore the original encoded data 2000. In this example, when an undo releasing instruction is input after the code line temporary truncation process, the encoded data 2004 is discarded, so that it becomes impossible to restore the

original encoded data 2000. It should be noted that after the code line discarding process, the code line temporary truncation process can be performed on the encoded data 2002 again.

**[0086]** Furthermore, when inputting the code line truncating instruction, it is also possible to instruct a progression order change of the encoded data in accordance with a necessity. When the instruction for the progression order change is provided, the code line temporary truncation process unit 1010 changes the progression order before performing the code line temporary truncation process.

**[0087]** The image processing apparatus has an operation mode (1-2) in which the same code line temporary truncation process as in the operation mode (1-1) is performed on all frames of a moving image in the form of Motion-JPEG2000. FIG. 11 is a flowchart of the operation mode (1-2).

**[0088]** At Step S151, the encoded data in the storage unit (A) 1004A is selected such that the encoded data of a beginning frame is first selected. At Step S152, the code line truncation command corresponding to a default code line truncating method or a code line truncating method specified by the user is issued from the control unit 1020, and the code line temporary process unit 1010 performs the code line temporary truncation process on the encoded data of the selected frame. Steps S151 and S152 are repeated, and when the last frame is processed (Yes in Step S153), this operation mode is terminated.

**[0089]** The image processing apparatus also has an operation mode (1-3) of restoring the encoded data of all frames processed by the code line temporary truncation process in the operation mode (1-2). A flowchart of this operation mode (1-3) is omitted, but an undo command may be issued from the control unit 1020, and under the control of the control unit 1020, the restoration process unit 1011 performs

the restoration process by the frame of the moving image.

[0090] The image processing apparatus also has an operation mode (1-4) of discarding the code lines of the all frames of the moving image that have been temporarily truncated in the operation mode (1-2). The flowchart of this operation mode (1-4) is omitted, but an undo releasing command is issued from the control unit 1020, and under the control of the control unit 1020, the encoded data in the storage unit (B) 1004B that consists of the truncated code lines of the frames is erased. After that, the original encoded data of the frames cannot be restored.

[0091] The image processing apparatus also has an operation mode (1-5) of temporarily cutting frames (performing a frame-by-frame code line temporary truncating) for encoded data of a moving image in the form of Motion-JPEG2000. FIG. 12 is a flowchart of this operation mode (1-5). The user previously selects the method of selecting frames to be temporarily cut. For example, as the method of selecting frames to be temporarily cut, there are a method of selecting every Nth ( $\geq 1$ ) frame, a method of selecting frames that are less different from the preceding frames (i.e., selecting a frame with less movement), a method of selecting frames corresponding to images having large camera vibration out of the moving image, and a method of selecting frames that do not include a person.

[0092] At Step S201, from a beginning frame, the control unit 1020 successively selects a frame out of a moving image whose encoded data is stored in the storage unit (A) 1004A, and determines whether or not the selected frame should be temporarily truncated at Step S202. For example, when the method of temporarily selecting every Nth frame is selected, the control unit 1020 makes this determination based on the frame number. When the method of temporarily selecting frames less different from the preceding frames (frames with less movement), or the method of



temporarily selecting large camera-shake frames is selected, the control unit 1020 makes this determination based on a analyzing result of the analyzing unit 1022. The operation of the analyzing unit 1022 will be described later.

**[0093]** At Step S203, the frame-by-frame code line temporary truncation command for the frame that is determined to be temporarily truncated is issued from the control unit 1020, and the frame-by-frame code line temporary truncation process is performed on the encoded data of this frame in the code line temporary truncation process unit 1010. Specifically, in the frame-by-frame code line temporary truncation process, the encoded data of the temporarily truncated (selected) frame is stored in the storage unit (B) 1004B, and the encoded data in the storage unit (A) 1004A corresponding to the selected frame is erased. That is, this frame is truncated from the group of frames of the moving image, but the encoded data of this truncated frame is stored. In this manner, the processes of Steps S201, S202, and S203 are repeated until the operation reaches the last frame. When the last frame is processed (Yes at Step S204), the procedure of this operation mode is terminated.

**[0094]** The image processing apparatus further has an operation mode (1-6) of reproducing a group of frames accumulated in the storage unit (A) 1004A. A flowchart of this operation mode (1-6) is omitted. However, in this operation mode (1-6), under the control of the control unit 1020, the encoded data of the frames in the group of frames of the moving image stored in the storage unit (A) 1004A is successively decoded and expanded so that the reproduced frame images can be displayed on the displaying unit 1014.

**[0095]** After the temporary frame cutting is performed on every Nth frame or the frames with less movement of the moving image in the operation mode (1-5), if the image processing apparatus is made to operate in the operation mode (1-6),

confirming the contents of the moving image requires an amount of time shorter than the amount of time required when the temporary frame cutting is not performed.

Particularly, in a case of a moving image acquired by a monitoring camera, by temporarily cutting frames with less movement or frames that do not include a person, the contents of the moving image can be confirmed effectively.

**[0096]** The image processing apparatus further has an operation mode (1-7) of restoring the original encoded data from a state where frames have been temporarily cut (i.e., including the encoded data of cut frames in the group of frames). A flowchart of this operation mode (1-7) is omitted. However, an undo command is issued from the control unit 1020, and under the control of the control unit 1020, the encoded data of the temporarily cut frames stored in the storage unit (B) 1004B is written in the storage unit (A) 1004A by the restoration process unit 1011, and the encoded data in the storage unit (B) 1004B corresponding to the encoded data written in the storage unit (A) 1004A is erased. After this restoration process is performed, it becomes possible to reproduce the original moving image by using the operation mode (1-6). Furthermore, after the restoration process is performed, another method of temporarily cutting frames can be specified to temporarily cut frames by using the operation mode (1-5).

**[0097]** As described above, after the code line temporary cutting is performed so that the contents of a moving image can be confirmed efficiently, it is possible to restore all frames of the moving image in a complete state.

**[0098]** Furthermore, in the operation mode (1-1), the temporarily cut frames can be specified one by one to include the specified frame in the group of frames. This is advantageous when the user wants to include the temporarily cut frame determined to be included in an important section of the moving image after the user

confirms the contents of the moving image.

**[0099]** FIG. 13 is an illustration of temporary cutting and restoring of frames of a moving image. In FIG. 13, the reference number 2100 designates a group of frames of the moving image before frames are temporarily cut. When the temporary frame cutting process is performed in the operation mode (1-5), for example, encoded data of frames having a mesh pattern in the group of frames 2100 is temporarily cut, and an encoded data line 2101 of the cut frames having the mesh pattern is stored in the storage unit (B) 1004B. An encoded data line 2102 of the frames that exclude the temporarily cut frames having the mesh pattern remains in the storage unit (A) 1004A. Thereafter, when the restoration process is performed in the operation mode (1-7), the encoded data line 2101 of the temporarily cut frames is included in (returned to) the storage unit (A) 1004A to restore the original group of frames 2100.

**[00100]** The image processing apparatus still further has an operation mode (1-8) of discarding the encoded data of all frames that are temporarily cut. A flowchart of this operation mode (1-8) is omitted. However, an undo releasing command is issued from the control unit 1020, and under the control of the control unit 1020, the encoded data of the frames that have been temporarily cut and stored in the storage unit (B) 1004B is erased by the code line discarding process unit 1012. After the code line discarding process is performed, it becomes impossible to restore the encoded data of the temporarily cut frames.

**[00101]** It should be noted that the code line temporary truncation process can be performed in the operation mode (1-2) for the moving image that has undergone the code line discarding process. That is, a combination of the frame cutting and the code line truncating of each frame can be applied to the moving image.

**[00102]** As another example, the code line temporary truncating may be

performed on all frames of a moving image in the operation mode (1-2), and the code line discarding process may then be performed in the operation mode (1-4).

Thereafter, the frame temporary cutting may be performed in the operation mode (1-5).

**[00103]** Next, the analyzing unit 1022 will be described. The analyzing unit 1022 includes an analyzing unit for identifying a frame with less movement, an analyzing unit for identifying a large camera-shake frame, and/or an analyzing unit for identifying a frame that does not include a person.

**[00104]** The analyzing unit for identifying a frame with less movement compares a code amount (excluding an amount for tag information) of encoded data of each current frame with a code amount of encoded data of the preceding frame when inputting the encoded data of the moving image. When a difference between the code amount of encoded data of a current frame and the code amount of the encoded data of the preceding frame is smaller than a threshold value, the current frame is determined as the frame with less movement. In a section including an acquired largely moving scene, a code amount largely changes between frames. On the other hand, in a section including an acquired less moving scene, a code amount less change between frames. A suitable method for estimating a code amount includes a method of estimating a code amount of the entire image region, a method of estimating a code amount of a specific image region (for example, a center part of the image), and a method of estimating a code amount of a specific component. Alternatively, in a case where image data of a moving image is input, the analyzing unit for identifying an image with less movement can analyze the input image data to detect moving between consecutive frames. The scope of the present invention also includes this operation.

**[00105]** The analyzing unit for identifying a large camera-shake frame determined a camera-shake degree of each frame based on wavelet coefficients obtained in a process of compressing and encoding each frame by the image compression unit 1003. For example, in a case of performing the two-dimensional wavelet transformation by the decomposition level of 3, sub-band coefficients as shown in FIG. 2D can be obtained. The coefficients of the sub-bands HL are high-frequency components in a vertical direction of the image, and the coefficients of the sub-bands LH are high-frequency components in a horizontal direction of the image. A high-frequency component amount  $Y_v$  used for a measure of the high-frequency component in the vertical direction is calculated by the following equation (1), using the coefficients of the sub-bands HL at respective levels shown in FIG. 2D. A high-frequency component amount  $Y_h$  used for a measure of the high-frequency component in the horizontal direction is calculated by the following equation (2), using the coefficients of the sub-bands LH at respective levels.

**[00106]** The equation (1) is:

$$Y_v = a_v \cdot \sum |1HL| + b_v \cdot \sum |2HL| + c_v \cdot \sum |3HL| .$$

**[00107]** The equation (2) is:

$$Y_h = a_h \cdot \sum |1LH| + b_h \cdot \sum |2LH| + c_h \cdot \sum |3LH| .$$

**[00108]** In the equations (1) and (2),  $a_v$ ,  $b_v$ ,  $c_v$ ,  $a_h$ ,  $b_h$ , and  $c_h$  are constants equal to or larger than 0.

**[00109]** In this example, the decomposition level number is 3. However, when the decomposition level number is larger or smaller than 3, the high-frequency component amounts can be calculated by using similar equations. Furthermore, the coefficients of the sub-bands at a part of levels (for example, only at the level 1) may be used. Additionally, the equations may be modified.

**[00110]** A main shooting object is generally located at a center part of the image,

and the other part or background of the image is often out-of-focus. Accordingly, only the wavelet coefficients corresponding to the center part of the image may be used for calculating the high-frequency component amounts. However, the coefficients corresponding to the entire image may be used for calculating the high-frequency component amounts.

**[00111]** The high-frequency component amounts calculated according to the above-described manner have a relation with camera vibration as shown in FIGS. 14A through 14D. When camera vibration does not exist, both of the high-frequency component amount  $Y_v$  in the vertical direction and the high-frequency component amount  $Y_h$  in the horizontal direction are large, and a difference between the high-frequency component amounts  $Y_v$  and  $Y_h$  is not so large, as shown in FIG. 14D. When camera vibration exists in the horizontal direction, the high-frequency component amount  $Y_h$  in the horizontal direction is greatly decreased compared with a case where camera vibration does not exist, as shown in FIG. 14A. When camera vibration exists in the vertical direction, the high-frequency component amount  $Y_v$  in the vertical direction is greatly decreased as shown in FIG. 14B. When camera vibration exists in an oblique direction, both of the high-frequency component amounts  $Y_v$  and  $Y_h$  are decreased as shown in FIG. 14C. By taking advantage of such nature of camera vibration, based on values of the high-frequency component amounts  $Y_v$  and  $Y_h$ , the analyzing unit for identifying a large camera-shake frame can determine whether camera vibration of each frame is large or small.

**[00112]** Furthermore, the similar relation for camera vibration is often recognized between code amounts of the sub-bands HL and LH of encoded data, so that the analyzing unit can perform the camera-shake determination based on such code amounts. The scope of the present invention includes this operation. When the

image acquiring unit 1002 includes a camera vibration detection sensor such as an accelerator sensor, the analyzing unit may identify a large camera-shake frame based on an output of the camera-shake detection sensor. The scope of the present invention includes this configuration.

**[00113]** The analyzing unit for identifying a frame that does not include a person perform a process of recognizing a person's face for image data of each frame input from the image acquiring unit 1021 or the interface unit 1001. At this time, the frame in which a person's face is not recognized is determined as a frame that does not include a person. In the case where the image acquiring unit 1021 detects a color temperature for automatic white balance adjustment, when the detected color temperature of the frame is close to a flesh color of a person, this frame may be determined as a frame that includes a person. When the detected color temperature of the frame is far away from the flesh color, this frame may be determined as a frame that does not include a person. The scope of the present invention includes this process and a configuration for this process.

**[00114]** An analyzing result of at least one of the above-described respective analyzing units may be stored in storage such as an inside memory of the analyzing unit 1022 or the control unit 1020, and/or the storage unit (B) 1004B.

**[00115]** Next, another embodiment of the present invention will be described.

**[00116]** In a case of a JPEG2000 code, packets are arranged in an order determined by the above-described nested for-loop corresponding to a progression order. Accordingly, for example, in a case of the LRCP progression, if tag information is rewritten such that a for-loop of a layer is executed only two times, packets of only the layers 0 and 1 are handled at the side of a decoder, so that the decoding is partially performed. Similarly, in a case of the RLCP progression, if tag

information is rewritten such that a for-loop of a resolution is executed only two times, packets of only the resolution levels 0 and 1 are decoded at the side of the decoder.

**[00117]** Thus, in the case of JPEG2000 code, by manipulating the operation of rewriting tag information, the decoding can be partially performed at the side of the decoder. In other words, although a part of the original code line is not actually truncated, the decoder is made to determine that the part of the original code line has been truncated.

**[00118]** Information required to be rewritten to accurately perform this partial decoding is the number of the uppermost elements in the progression order, and the related information. The uppermost element is an element of a for-loop that is at the most outer side out of nested for-loops. For example, in the case of the LRCP progression, the uppermost element is a layer, and in the case of the RLCP progression, the uppermost element is a resolution.

**[00119]** In the second embodiment, the code line temporary truncation process is performed by rewriting tag information. The decoding process is performed by returning the rewritten tag information into the tag information having the contents before the rewriting. The code line discarding process is performed such that truncating target code lines are actually truncated and discarded from the encoded data after the code line temporary truncating is performed.

**[00120]** In the following, an outline of tag information added to encoded data of JPEG2000 will be described within an exemplary range. FIG. 15 shows a configuration of a main header. Marker segments of SIZ, COD and QCD are essential segments, but the other marker segments are optional segments.

**[00121]** FIGS. 16A and 16B show a configuration of a tile header. FIG. 16A



shows a header added to a head of tile data, and FIG. 16B shows a header added to a head of a divided partial tile line when a tile is divided into a plurality of parts. There is no essential tile marker segment in the tile header, and all tile marker segments are optional.

**[00122]** FIG. 17 shows a list of markers and marker segments. A configuration of an SIZ marker segment is shown in FIG. 18, a configuration of a COD marker segment is shown in FIG. 19, a configuration of a COC marker segment is shown in FIG. 20, a configuration of a QCD marker segment is shown in FIG. 21, a configuration of a QCC marker segment is shown in FIG. 22, and a configuration of a COM marker segment is shown in FIG. 23.

**[00123]** Following are several concrete examples (a) through (d) of rewriting tag information in the code line temporary truncation process.

**[00124]** In the example (a), the code line temporary truncation process is performed by the layer in encoded data of the LRCP progression. In this example, the tag information to be rewritten is the number of layers of SGcod of the COD marker segment shown in FIG. 19.

**[00125]** In the example (b), the code line temporary truncating is performed by the resolution level in encoded data of the RLCP or RPCL progression. In this example, tag information to be rewritten is Xsiz, Ysiz, XTsiz, YTsiz of the SIZ marker segment shown in FIG. 18, the decomposition level number of SPcod of the COD marker segment shown in FIG. 19, and Lqcd and SPqcd of the QCD marker segment shown in FIG. 21. In addition to that, when the COC marker segment shown in FIG. 20 exists, the decomposition level number of SPcoc of the COC marker segment is also rewritten. Further, when the QCC marker segment shown in FIG. 22 exists, the decomposition level number of SPqcc of the QCC marker segment is also

rewritten.

**[00126]** In the example (c), the code line temporary truncating is performed by the tile in encoded data in an arbitrary progression order. In this example, tag information to be rewritten is Xsiz and Ysiz of the SIZ marker segment shown in FIG. 18.

**[00127]** In the example (d), the code line temporary truncating is performed by the component in encoded data of the CPRL progression. In this example, tag information to be rewritten is Lsiz, Csiz, Ssiz, XRsiz, and YRsiz of the SIZ marker segment shown in FIG. 18. Furthermore, when the component transformation for SGcod of the COD marker segment shown in FIG. 19 includes "1", the component transformation "1" is changed to "0" because JPEG2000 specified that the component transformation is performed on only the first three components. For example, in JPEG2000, when the first three components out of four components are intended to be left, each component transformation may be "1," but when only one or two components out of four components are intended to be left, the component transformation corresponding to truncation process needs to be changed to "0."

**[00128]** Also, in one embodiment, the code line temporary truncating (temporary frame cutting) can be performed by the frame on encoded data of a moving image of Motion-JPEG2000. In this case, information for identifying temporarily cut frames may be added to these frames, but these frames are not actually cut from a group of frames of the moving image. In this example, a comment to the effect that the frame has been temporarily cut is described as Ccom of the COM marker segment (in FIG. 23) in the main header of the encoded data of the frame to be temporarily cut. The restoration process may be performed such that this comment showing that the frame has been temporarily cut is discarded. The information for

identifying the temporarily cut frame may be added to the frame in any form if the information may be identified in the image processing apparatus. For example, the SIZ marker is rewritten in an ineffective bit line, and the restoration process is performed such that the SIZ marker is returned from the ineffective bit line to an effective bit line.

**[00129]** Next, a configuration and operation of the image processing apparatus according to another embodiment of the present invention will be described.

**[00130]** The configuration of the image processing apparatus in this embodiment is partially different from that of the previous embodiment, so the configuration in this embodiment is described with reference to FIG. 7. In this embodiment, a code line temporary truncation process unit 1010 is adapted to perform a code line temporary truncation process on a static image or each frame of a moving image by the above-described rewriting of tag information, and to perform a temporary frame cutting by adding the above-described comment. A restoration process unit 1011 is adapted to perform a restoration process on a static image or individual frame images by turning the rewritten tag information into the original tag information, and to perform a restoration process on the temporary cut frames of the moving image by discarding the comments. A code line discarding process unit 1012 is adapted to actually discard a truncating target code line from encoded data of a static image or individual frame images after the encoded data undergoes the code line temporary truncating. Further, the code line discarding process unit 1012 is adapted to create encoded data in which a truncating target code line has been truncated from the original code line, and to discard encoded data of the temporarily cut frames. In one embodiment, it is not necessary to separately use two storage units (storage areas) 1004A and 1004B, so that the storage units 1004A and 1004B together

will be called a storage unit 1004. The other units may be the same as in the first embodiment.

**[00131]** The image processing apparatus according to one embodiment has operation modes (2-1) through (2-8) corresponding to the operation modes (1-1) through (1-8) of the initial embodiment described above. The operation modes (2-1) through (2-8) will be described sequentially.

**[00132]** The operation mode (2-1) will be described with reference to FIG. 8. In the operation mode (2-1), first, the user specifies one image, and the code line temporary truncation process is performed on encoded data of the specified image. The user can confirm a result of this code line temporary truncation process.

**[00133]** When an image (for example, a static image or a frame of a moving image) is specified, at Step S101, under the control of the control unit 1020, encoded data of the specified image is read from the storage unit 1004, and is input to the image expanding unit 1013 so that the decoded and expanded image data can be displayed on the displaying unit 1014.

**[00134]** When the user watches the displayed image, and determines that a code line (a part of the original code line) needs to be truncated from the original code line, the user specifies a code line truncating method, and inputs a code line truncating instruction. The specifying the code line truncating method is the same as in the first embodiment.

**[00135]** The control unit 1020 determines the contents of user's input at Step S102, and performs control in accordance with the contents of the input.

**[00136]** When the input by the user is a code line truncating instruction, a code line truncation command that includes information about the code line truncating method specified by the user is issued from the control unit 1020 to the code line

temporary truncation process unit 1010, and under the control of the control unit 1020, the code line temporary process is performed by the code line temporary truncation process unit 1010 such that tag information of the encoded data is rewritten in accordance with the specified code line truncating method at Step S103.

Subsequently, under the control of the control unit 1020, the encoded data in which the tag information has been rewritten is read from the storage unit 1004, and is decoded and expanded in the image expanding unit 1013, and image data of the decoded and expanded data is displayed on the displaying unit 1014 at Step S104. At this time, the tag information has been rewritten so that the truncating target code line is not decoded and expanded. In this manner, by rewriting the tag information, it is possible to reproduce the same image as in a case where the truncating target code line is actually truncated.

**[00137]** The user confirms quality or the like of the displayed image. When the user determines that a code line truncating should be performed over again, the user can input an undo instruction. On the other hand, when the user determines that a result of the current code line truncating should be fixed, the user can input an undo releasing instruction, and when the user desires to terminate the operation, the user can input an end instruction.

**[00138]** When an undo instruction is input, an undo command is issued from the control unit 1020 to the restoration process unit 1011, and the restoration process is performed by the restoration process unit 1011 such that the tag information of the encoded data that has been rewritten by the code line temporary truncation process is returned to the original contents at Step S105. Subsequently, under the control of the control unit 1020, the restored encoded data is read from the storage unit 1004, and is decoded and expanded in the image expanding unit 1013, and the decoded and

expanded image is displayed on the displaying unit 1014 at Step S106. For the encoded data of the displayed image, the user can input a code line truncating instruction with a different code line truncating method again so that a code line temporary truncation process can be performed in the different method.

[00139] Meanwhile, when an undo releasing instruction is input by the user, an undo releasing command is issued from the control unit 1020 to the code line discarding process unit 1012, and a code line discarding process is performed by the code line discarding process unit 1012 at Step S107. In this code line discarding process, the truncating target code line of the encoded data that has undergone the code line temporary truncation process is actually discarded (erased), and only the encoded data after this code line discarding process is stored in the storage unit 1004. As a result, the original data having the truncating target code line can no longer be restored.

[00140] After the code line temporary truncation process is performed, the user can input the end instruction to terminate the operation, and at another time, input an undo instruction or an undo releasing instruction for the encoded data by using the operation mode (2-1) so that a restoration process or a code line discarding process can be performed.

[00141] When inputting a code line truncating instruction, the user can instruct a change of a progression order of the encoded data in accordance with a necessity. When this instruction for the progression order change is provided, the code line temporary truncation process unit 1010 changes the progression order, and then performs a code line temporary truncation process in the changed progression order.

[00142] Next, the operation mode (2-2) in the case where a code line temporary truncation process similar to the operation mode (2-1) is performed on all frames of a

moving image of Motion-JPEG2000 will be described with reference to FIG. 11.

**[00143]** A frame of encoded data in the storage unit 1004 is successively selected from a beginning frame by the control unit 1020 at Step S151. A code line truncation command with a code line truncating method specified by the user or a default code line truncating method is issued from the control unit 1020, and the code line temporary truncation process unit 1010 performs a code line temporary truncation process on encoded data of the selected frame such that tag information of the encoded data is rewritten at Step S151. The same process is repeated until the last frame is processed. When the last frame is processed (Yes at Step S153), the operation of this operation mode (2-2) is terminated.

**[00144]** The image processing apparatus has the operation mode (2-3) of restoring the encoded data of all frames that have undergone the code line temporary truncation process in the operation mode (2-2). A flowchart of the operation mode (2-2) is omitted. However, in the operation mode (2-3), an undo command is issued from the control unit 1020, and under the control of the control unit 1020, the frames of the moving image are successively restored by the restoration process unit 1011 such that the tag information is turned into the original information.

**[00145]** The image processing apparatus also has the operation mode (2-4) of performing a code line discarding process on all frames of the moving image that have undergone the code line temporary truncation process in the operation mode (2-2). A flowchart of the operation mode (2-4) is omitted. However, in the operation mode (2-4), an undo releasing command is issued from the control unit 1020, and under the control of the control unit 1020, the truncating target code line of the encoded data of each frame in the storage unit 1004 is discarded by the code line discarding process unit 1012.

**[00146]** Next, the operation mode (2-5) in which frame temporary cutting is performed on encoded data of a moving image in the form of Motion-JPEG2000 will be described with reference to FIG. 12. A user can previously specify a frame temporary cutting method. The frame temporary cutting method is the same as in the previously described embodiment.

**[00147]** At Step S201, from a beginning frame, the control unit 1020 successively selects a frame of a moving image whose encoded image is stored in the storage unit 1004, and determines whether or not the current selected frame should be truncated at Step S202. For example, when a method of temporarily cutting every Nth frame is specified, the control unit 1020 makes this determination based on the frame number. When a method of temporarily cutting a frame less different from the preceding frame or a method of temporarily cutting a frame having large camera vibration is specified, the control unit makes this determination based on an analyzing result of the analyzing unit 1022.

**[00148]** For the frame that is determined to be temporarily cut, a frame temporary cutting command is issued from the control unit 1020, and a comment to the effect that the frame has been temporarily cut is written in a COM marker segment of a main header of the encoded data of this frame by the code line temporary truncation process unit 1010 at Step S203. Steps S201, S202 and S203 are repeated until the last frame of the moving image is processed. When the last frame is processed (Yes at Step S204, this operation mode is terminated.

**[00149]** Next, the operation mode (2-6) of reproducing a moving image whose encoded data is accumulated in the storage unit 1004 will be described with reference to FIG. 24. FIG. 24 is a flowchart of the operation mode (2-6).

**[00150]** Encoded data of one frame is selected by the control unit 1020



successively from a beginning frame at Step S301, and the contents of a COM marker segment of a main header of the selected frame is checked at Step S302. When the comment to the effect that the frame has been temporarily cut is described in the COM marker segment (Yes at Step S302), this frame is not reproduced, and the next frame is selected. On the other hand, when the comment to the effect that the frame has been temporarily cut is not described in the COM marker segment, the encoded data of this frame is read by the image expanding unit 1013 where the encoded data of this frame is decoded and expanded, and an image of the decoded and expanded data is displayed on the displaying unit 1014 at Step S303. Step S301, S302, and S303 are repeated until the last frame is processed. When the last frame is processed (Yes at Step S304), this operation mode (2-6) is terminated.

**[00151]** For example, when frames having a mesh pattern out of the group of frames 2100 shown in FIG. 13 have been temporarily cut, only the group of frames 2102 shown in FIG. 13 is successively reproduced. As a result, it is possible to effectively confirm the contents of the moving image in a short time compared to a case where frames are not temporarily cut.

**[00152]** The image processing apparatus also has the operation mode (2-7) of restoring the encoded data of the temporarily cut frames. A flowchart of the operation mode (2-7) is omitted. However, in this operation mode (2-7), an undo command is issued from the control unit 102, and under the control of the control unit 1020, the encoded data of the temporarily cut frames is successively read from the storage unit 1004 by the restoration process unit 1011, and the comment that is used for identifying the temporarily cut frame and that is described in the COM marker segment of the frame is discarded. After this restoration process is performed, the original moving image can be reproduced by using the operation mode (2-6).

Thereafter, by using the operation mode (2-1), another temporary frame cutting method may be specified so that the temporary frame cutting can be performed in this method.

**[00153]** The image processing apparatus also has the operation mode (2-8) of discarding encoded data of all frames that have been temporarily cut. A flowchart of the operation mode (2-8) is omitted. However, in this operation mode (2-8), an undo releasing command is issued from the control unit 1020, and under the control of the control unit 1020, the encoded data of respective temporarily cut frames in the storage unit 1004 is successively erased by the code line discarding process unit 1012. After this code line discarding process is performed, it becomes impossible to restore the temporarily cut frames.

**[00154]** Moreover, the code line temporary truncating in the operation mode (2-2) can be performed on the moving image that has undergone this code line discarding process. That is, a combination of the temporary frame cutting and the code line temporary truncating for each frame can be applied to the moving image.

**[00155]** Alternatively, the code line temporary truncating may be performed on each frame of the moving image in the operation mode (2-2), the code line discarding process may then be performed in the operation mode (2-4), and thereafter, the temporary frame cutting may be performed in the operation mode (2-5).

**[00156]** As understood from the above description, according to the present invention, by using the code line temporary truncation process and the restoration process, it is possible to repeatedly attempt code amount truncating of encoded data of a static image or frames of a moving image. Since the code line temporary truncation process and the restoration process do not accompany recompression of encoded data, even if code amount decreasing is repeatedly performed, image

degradation caused by a generation does not occur. Furthermore, when restoring of original encoded data is not needed, unnecessary code line or frames of a moving image can be discarded by the code line discarding process, so that it is possible to effectively use a storing medium that stores encoded data. In addition, since the frame temporary truncation process can decrease the number of frames of a moving image to be reproduced, it is possible to confirm the contents of the moving image in a short time. Particularly, the frame temporary cutting process can be automatically performed on frames that do not include a person or frames with less movement of a moving image, so that it is possible to very efficiently confirm the contents of a moving image captured by a monitoring camera. Even after the frame temporary cutting process is performed, the original moving image can be completely restored by the restoration process, so that it is possible to perform the frame temporary cutting process many times.

**[00157]** This patent application is based on Japanese priority patent application No. 2002-261739 filed on September 6, 2002, the entire contents of which are hereby incorporated by reference.